

Primer and Recent Developments on Fountain Codes

Jalaluddin Qureshi*, Chuan Heng Foh†, Jianfei Cai*

* School of Computer Engineering

Nanyang Technological University, Singapore

† Centre for Communication Systems Research

Department of Electronic Engineering

University of Surrey

Surrey GU2 7XH, United Kingdom

Abstract—In this paper we survey the various erasure codes which had been proposed and patented recently, and along the survey we provide introductory tutorial on many of the essential concepts and readings in erasure and Fountain codes. Packet erasure is a fundamental characteristic inherent in data storage and data transmission system. Traditionally replication/retransmission based techniques had been employed to deal with packet erasures in such systems. While the Reed-Solomon (RS) erasure codes had been known for quite some time to improve system reliability and reduce data redundancy, the high decoding computation cost of RS codes has offset wider implementation of RS codes. However recent exponential growth in data traffic and demand for larger data storage capacity has simmered interest in erasure codes. Recent results have shown promising results to address the decoding computation complexity and redundancy tradeoff inherent in erasure codes.

I. INTRODUCTION

The aim of this paper is to provide a self-contained introduction about the erasure coding scheme and recent developments in Fountain codes, which are currently the predominant class of erasure codes.

Packet erasure is one the fundamental and inevitable characteristic in data transmission and data storage system. For example routers may drop a packet due to congestion. Similarly a file in a data storage system can be erased due to component failures. The problem of packet erasure exuberates for data transmission on wireless channel due to the shared medium of transmission resulting in packet collisions. In addition to packet collision, for wireless channel, packet may also be erased due to channel fading, additive white Gaussian noise (AWGN) and signal attenuation. The average wireless channel erasure rate in some deployments can be as high as 20-50% [1], [2].

Traditional approach of dealing with packet erasure is to use replication and retransmission. The method of replication and retransmission introduces control overhead. For data storage system, replication provides limited reliability. For instance in the event that the original file and the replicated files are both erased, then the data storage system can not recover the original file. Similarly the use of retransmission technique for

data transmission system is dependent on packet acknowledgement control frame from the client. It is also possible that an acknowledgement frame can also be erased due to the same reasons as the original data packet, erroneously resulting in the retransmission of those data packets which the client has already received.

For wireless networks, the transmission of acknowledgement frame occupies the wireless channel medium and therefore adversely affects the transmission bandwidth. The problem of collecting acknowledgement exacerbates when the transmitter is multicasting the data to N clients, in which case it has to collect N acknowledgement frames. Due to the exacerbation to efficiently collect ACK frames for multicast transmission, and efficiently retransmit the erased packets, legacy IEEE 802.11 multicast transmission is a “no-ACK, no-retransmission” scheme, in which the access point (AP) transmits the data packet and then waits for the channel to be free, conforming only to the carrier sense multiple access collision avoidance - (CSMA/CA) access procedure, before transmitting the next data packet.

The urgency to have a reliable multicast transmission scheme for 802.11 wireless networks is reflected by the recent formation of the IEEE working group, called Task Group aa (TGaa) [3]. The wireless multicast technology is also employed by Multimedia Broadcast Multicast Service (MBMS) for 3GPP cellular network (3GPP TS 26.346).

Erasure codes have been proposed as an efficient remedy to improve the reliability and scalability of data transmission over erasure channels. In an erasure codes transmission scheme the client can recover the k data packets from the n transmitted packets, where $n = (1 + \epsilon)k$, at a rate r given as n/k , and ϵ is known as the *overhead* (or *redundancy*) of the coding scheme. For recovering the input packets, it is irrelevant which packets the client had received, as long as it has received any k linearly independent packets, the client can decode the k input packets. In erasure coding, the coded packets are generated by linearly mapping the packets with coefficients from a finite field \mathbb{F}_q , where for computer science applications the *finite field size* q is given as $q = 2^i$, $i \in \mathbb{N}_1$, and \mathbb{N}_1 is the set of natural numbers excluding zero. Erasure codes, such as Triangular codes, may also be non-linear, and therefore may involve operations over

real field.

Erasure coding is a technique to provide reliability in an event of packet erasure. Error coding and erasure coding are class of *Forward Error Correction* (FEC) techniques. *Error coding* scheme protects the system from data corruption, e.g. noise or attenuation, whereas *erasure coding* protect the system from data lost during transmission, e.g. packet drop at router due to congestion or packet collision over the wireless channel. In our paper we only consider a *binary erasure channel* (BEC) model. In a BEC a transmitted packet is either correctly received with probability p , or not received with probability $1 - p$. Similarly a stored data packet is either not corrupted with probability p , or corrupted with probability $1 - p$. The BEC is also known as Bernoulli channel model. When considering data transmission to multiple clients, i.e. multicast or broadcast, the packet erasure probability on each of the transmission channel is assumed to *independent and identically distributed* (iid).

To illustrate how erasure coding can be improve the reliability of a transmission system, consider for example a wireless network where an AP is multicasting packets c_1 and c_2 to clients R_1 and R_2 . However, R_1 receives c_1 but not c_2 , whereas R_2 receives c_2 but not c_1 . In this case, rather than retransmitting packet c_1 and c_2 in two different time slots, it is possible for the transmitter to encode the packets $c_1 \oplus c_2$ over \mathbb{F}_2 , and transmit the encoded packet in one time slot. On receiving the encoded packet both the client can recover the lost packet by decoding the original packet with the encoded packet. This therefore reduces the number of retransmissions from two rounds to one, and hence improves the network bandwidth.

Similarly erasure coding can also improve the reliability performance in data storage system. Consider for example packet c_1 and c_2 being stored on a data storage system. To improve the reliability of the system, traditional approach would replicate c_1 and c_2 , and store two copies of c_1 and two copies of c_2 . However in an event of storage failure, where two copies are erased, and both these erased copies happen to be c_1 . In this case there is no way the system can recover packet c_1 . On the contrary, if instead of storing the replicated packet, the system stores two coded packets $c_1 + c_2$ and $c_1 + \alpha c_2$, then in an event of storage failure, where two packets are erased, then irrespective of which these two erased packets are, the system can still recover c_1 and c_2 from the remaining two packets.

Such reliability gains however do not come without trade-offs. While linear coding schemes over larger field size such as the random linear (RL) codes and Reed-Solomon (RS) codes can deliver optimal rate, decoding RS and RL coded packets are costly, requiring the use of matrix inversion which is implemented using Gaussian elimination with complexity $\mathcal{O}(k^3)$ when the coding coefficients are dense, or variants of the Wiedemann algorithm with complexity $\mathcal{O}(k^2 \log k)$ when

the coding coefficients are sparse [4], in addition to the cost of matrix multiplication.

It has been further shown that the decoding computation cost is also dependent on the field size from which the coding coefficients are selected. Practical implementation of RL codes on iPhone 3G has shown that the decoding throughput of RL codes over \mathbb{F}_2 is approximately six times faster than decoding RL codes over \mathbb{F}_{256} on the same testbed. Similarly encoding over \mathbb{F}_2 is approximately eight times faster compared to encoding over \mathbb{F}_{256} [5]. Unfortunately smaller field size can not be used for RL codes, as larger field size is a prerequisite for RL codes to deliver optimal rate.

Experimental evaluation of RL codes over \mathbb{F}_{256} on iPhone 3G, for $k = 64$ with packet length of 4096 bytes, has shown that for two devices with same configurations and running the same applications, the device running with an additional RL decoding application consumes approximately 20% more battery energy reserves [6]. Mobile phone batteries suffer from severe energy limitation, which is why handset vendors are increasingly interested in energy optimization of various smartphone applications which can sustain longer operational time.

The high decoding cost of packets coded over large field size can be addressed by using the simpler XOR addition for encoding and decoding, which is also known as \mathbb{F}_2 addition. It has been shown that XOR addition of two packets, each 1000 bytes long only consumes 191 nJ of energy [7]. Given that the transmission of a packet of the same length over IEEE 802.11 network on Nokia N95 consumes 2.31 mJ of energy [7], the overall energy cost of XOR addition has no apparent affect on the operational time of a mobile phone.

It is apparent from the above discussion, that an amicable solution is needed to address the throughput performance and decoding computation cost tradeoff inherent in erasure coding. In the last decade a subclass of erasure codes, known as Fountain codes have gained widespread acceptance for its ability to address this tradeoff. However Fountain codes assume very large input packet length to deliver near-optimal transmission rate using linear decoding algorithm. For example, even for very large input packet length of $k \approx 10,000$, LT codes, which is an implementation of Fountain codes, have an overhead of about 5% [8]. Similarly for $k = 65,536$, Raptor codes, which is also an implementation of Fountain codes, have an overhead of about 3.8% [9, Table 1].

More recently, Qureshi *et al.* proposed the Triangular coding scheme [10] to address the tradeoffs in performance, computation costs and packet length in erasure codes. An ideal erasure code should be able to deliver near-optimal transmission rate with linear computation cost, with such performance being independent of any parameter including input packet length. As we will show, the Triangular codes have the potential to be designed to fulfill the characteristics of such ideal codes.

The rest of the paper is organized as follow. We first present a tutorial on erasure codes in Section II, and the characteristics of classical erasure coding schemes, the Reed-Solomon codes, low density parity check codes, and random

Strictly speaking, RL codes are suboptimal, however the difference between optimal rate and the rate of RL codes over large field size $\mathbb{F}_{q \geq 256}$ is negligible, and in order of $\approx 10^{-4}$.

linear codes in Section III. We then present the highlight and performance of Fountain codes, namely Tornado codes, Luby-Transform codes, Raptor codes and standardized Raptor codes in Section IV, and those of Triangular codes in Section V. We then conclude with open research problems in erasure coding in Section VI and summary of our work in Section VII.

II. TECHNICAL BACKGROUND

We provide a brief mathematical background on the encoding and decoding for linear codes that covers Fountain codes. Let there be k numbers of input data packets to be encoded for transmission. The set of k input data packets is given by the vector $\mathbf{M} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k]$, and the set of encoded packets transmitted by the server is given by the vector $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j]$. If $\mathbf{M} \subseteq \mathbf{X}$, then such a coding scheme is known as *systematic codes*, and *non-systematic codes* otherwise. The set of innovative packets which the client has received is given as \mathbf{Y}_i , $\mathbf{Y}_i \subseteq \mathbf{X}$. We define a packet as innovative packet, if the packet is linearly independent with respect to all the packets that a client already has.

A. Linear Scalar Codes

A linear code \mathcal{C} of length B over the finite field \mathbb{F}_q is formally defined as a linear subspace of the vector space \mathbb{F}_q^B , and *non-linear* otherwise. The code \mathcal{C} , $\mathcal{C} \subseteq \mathbb{F}_q^B$, is a linear subspace of \mathbb{F}_q^B , if for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}$, $\mathbf{x}_i \oplus \mathbf{x}_j \in \mathcal{C}$ and for all $\mathbf{g}_m \in \mathbb{F}_q$, $\mathbf{x}_i \in \mathcal{C}$, $\mathbf{g}_m \cdot \mathbf{x}_i \in \mathcal{C}$. Therefore the subset code $\mathcal{D} = \{111, 010\}$ of \mathbb{F}_2^3 is not a linear code, as $111 \oplus 010 = 101 \notin \mathcal{D}$.

In this paper, we only consider scalar erasure codes. In *scalar coding*, a packet can not be split into smaller sub-packets, whereas in *vector coding*, a packet may be split into smaller sub-packets. Vector codes are shown to outperform scalar codes in certain classes of transmission problem using erasure codes such as the cooperative data exchange problem [11].

B. Packet Encoding

To encode a packet, the server chooses coefficients vectors from a finite field \mathbb{F}_q , to form an encoding coefficient vector $\mathbf{G}_j = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k]$, $\mathbf{g}_k \in \mathbb{F}_q$, which is then multiplied with vector \mathbf{M} to generate a single coded packet \mathbf{x}_j given as,

$$\mathbf{x}_j = \mathbf{G}_j \mathbf{M}^T. \quad (1)$$

Coding over field size $\mathbb{F}_{q>2}$, requires finite field multiplication and XOR addition operations. Whereas for XOR coding, $\mathbf{g}_k \in \{0, 1\}$, only the addition operation is required. For the client to be able to decode the coded packets, the transmitter needs to add information about the vector \mathbf{G}_j in the packet header. The number of bits required to store one such coefficient \mathbf{g}_k is given as q bits. Since there are k such coefficients, the packet overhead for a coded packet is given as $k \times q$ bits.

C. Packet Decoding

After a client R_i has received k innovative packets, these k packets are placed in matrix \mathbf{Y}_i . The coding coefficients from all the coded packet's header is used to form a $k \times k$ coefficient matrix \mathbf{H}_i . The set of original packets \mathbf{k} can then be decoded by R_i as,

$$\mathbf{M}^T = \mathbf{H}_i^{-1} \mathbf{Y}_i^T. \quad (2)$$

Inversion of \mathbf{H}_i can be performed using Gaussian elimination. We illustrate the above decoding process with the aid of simple example. Consider the three received coded packets given as, $x_1 = c_1 \oplus c_2$, $x_2 = c_2 \oplus c_3$ and $x_3 = c_1 \oplus c_2 \oplus c_3$. The corresponding \mathbb{F}_2 matrix \mathbf{H}_i , and its inverse, are given as,

$$\mathbf{H}_i = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{H}_i^{-1} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

One can verify that based on the inverted matrix, the decoded packets are given as, $c_1 = x_2 \oplus x_3$, $c_2 = x_1 \oplus x_2 \oplus x_3$, and $c_3 = x_1 \oplus x_3$.

Coefficient matrix \mathbf{H}_i for XOR coded packets will form a \mathbb{F}_2 matrix, and only require the "row interchange" and "row addition" operations [12] for Gaussian elimination. After the inversion of the binary matrix, only the addition operation needs to be performed. However for packets coded over larger field size, Gaussian elimination process would also need to perform "row scaling" operation (i.e. multiplying a row of the matrix with a non-zero scalar) in addition to the "row interchange" and "row addition" operations. After the inversion, the client will need to perform multiplication and addition to decode the native packets. Therefore even though the complexity order of matrix inversion for both \mathbb{F}_2 matrix and $\mathbb{F}_{q>2}$ is same, decoding XOR coded packets requires fewer computation steps.

D. Decoding Algorithms

All linear erasure codes essentially use the Gaussian elimination or one of its variants to perform decoding. Gaussian elimination consists of two major steps, *triangularization* of the matrix into an upper or lower Triangular matrix, with complexity $\mathcal{O}(k^3)$ for a $k \times k$ full rank matrix, and *back-substitution* of the triangular matrix to solve the unknown variables with complexity $\mathcal{O}(k^2)$ [12]. The belief-propagation (BP) decoding algorithm used for LT codes, and the inactivation decoding algorithm used for Raptor codes [13, pp. 247-255], can therefore be entirely described using Gaussian elimination steps of triangularization and back-substitution. We refer interested readers to [4, Chapter 11] and the references therein for discussion on various methods to solve a system of linear equations.

When the Triangular matrix has an average sparsity of ω , then the complexity of back-substitution is given as $\mathcal{O}(k\omega)$. It is interesting to note that the average sparsity ω , of a Triangular matrix is bounded as $\frac{k+1}{2}$, which explains the derivation of total number of computation steps for back substitution as

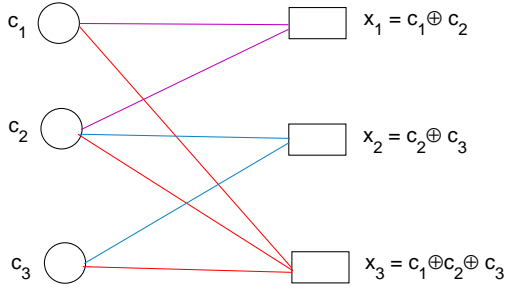


Fig. 1. An example of an irregular Tanner graph representing the irregular XOR codes given in Section II-C.

$\frac{k^2}{2} + \frac{k}{2}$ in [12], i.e. k times ω . All efficient erasure coding scheme are essentially build on this concept of generating sparse coding coefficients, i.e. $\omega < \frac{k+1}{2}$, which can be solved using back-substitution. For LT and Raptor codes, ω is given by $\log(\frac{k}{\delta})$ and $\log(\frac{1}{\epsilon})$ respectively.

The work of Darmohray and Brooks [14] published in 1987 has shown the performance merits of Gauss-Jordan elimination over Gaussian elimination in multi-processors system despite the former higher computation cost. Gauss-Jordan elimination method in a multi-processor system still continues to be of significant implementation importance in recent works [15]. While the lower computational complexity of Wiedemann algorithm to solve sparse system of linear equations has motivated the design of sparse coding vector [16], over $\mathbb{F}_{q>2}$.

E. Tanner Graph

Many of the characteristics of Fountain codes are illustrated using the Tanner graph, which is a special class of the bipartite graph. In a Tanner graph, the input packets and coded packets are arranged into two disjoint vertex sets, i.e. there can only exist an edge between vertices represented by an input packet and coded packet. The set of packets used to encode the coded packet is given by the edges connecting the coded packet, as each of these edges will be connected to an input packet. The Tanner graph can be treated as a pictorial representation of a \mathbb{F}_2 matrix \mathbf{H}_i . An example of a Tanner graph is given in Figure 1.

A Tanner graph, and consecutively the codes that the Tanner graph represents, is said to be a regular graph (/regular code) if each of the coded packet is generated using a fixed number of input packets, and irregular graph (/irregular code) otherwise.

F. Finite Field Arithmetic

Packet encoding and decoding represented by Equation (1)-(2) are done using finite field arithmetic, which is different from the real field arithmetic. Fundamentally finite field arithmetic stipulates that for encoding and decoding over a field

size \mathbb{F}_q , the operands are given as $\mathbb{F}_q = \{0, 1, \dots, q-1\}$, and the solution when these operands are added, subtracted, multiplied or divided, is also an element of the field \mathbb{F}_q . Because of this, multiplication in Equation (1) does not increases the length of packet, which would have otherwise been observed for real field multiplication.

The monograph by Vasilenko, translated by Martsinkovsky [4], covers many of the number-theoretic algorithms over finite field in an easy-to-understand writing style.

G. Related Problem

The erasure coding model assumes a single server broadcasting data to multiple clients over BEC, where all the clients want all the data being broadcast by the server. A coding scheme related to erasure code is the index code to solve the index coding problem [17].

The *index coding problem* is an instance of packet transmission problem to a set of N clients $R = \{r_1, r_2, \dots, r_N\}$ by the server having a set of k packets $P = \{c_1, c_2, \dots, c_k\}$, and the *side information* about the set of packets each client $r_i \in R$ has, $H_i \subset P$, and the set of symbol each client wants, $W_i \subseteq P \setminus H_i$, such that the total number of symbols transmitted is minimized.

The erasure codes can therefore be treated as a special case of index codes, where $W_i = P$, $\forall i$, at the start of the transmission. Finding optimal index codes and approximation to such optimal index codes are both NP-hard for the general index coding problem.

Another related problem is that of *network coding*, where multiple sources are multicasting data to multiple common clients connected to the sources through intermediate routers, with BEC model. In network coding, apart from the sources, intermediate routers can also perform encoding before forwarding the coded packets. For a self-contained introductory tutorial on network coding we refer readers to [18].

III. CLASSICAL ERASURE CODES

A. Reed-Solomon Codes

Some of the first class of erasure codes were based on the Reed-Solomon (RS) error coding scheme proposed by Reed and Solomon in 1960 as an error coding [19] scheme, and presented as an erasure coding scheme by Rizzo in 1997 [20]. The RS codes have also been suggested for use in hybrid Automatic Repeat reQuest (ARQ) protocol [21, Chap 7] to improve transmission reliability in wireless network.

RS code are linear codes, using coding coefficients \mathbf{G}_j from the Vandermonde matrix, M_n where the rows are given by a geometric progression sequence. Any k arbitrary rows from a $n \times k$ Vandermonde matrix, given as M'_k , form a nonsingular matrix, provided that the common ratio of each of the geometric progression sequence is unique. The client can decode the input packets once it has received any k coded packets from n transmissions, where $n \geq k$.

A Vandermonde matrix is given as follow,

$$M_n = \begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{k-2} & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{k-2} & a_2^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & a_n & a_n^2 & \dots & a_n^{k-2} & a_n^{k-1} \end{pmatrix},$$

Theorem 1. *The matrix M'_k is invertible if each of the number a_ℓ is unique.*

Proof: A proof of this can be found in [22, pp. 17-18]. ■

The main shortcoming of the RS codes is that decoding of RS codes requires the computationally expensive Gaussian elimination method, therefore limiting the size of the input packets. Since the RS codes use dense coding coefficients, computationally efficient algorithms to inverse the matrix such as structured Gaussian elimination and Wiedemann algorithms are of no help, as these algorithms run on sparse matrices.

Despite such shortcoming, RS codes still continues to be used for various applications such as redundant arrays of inexpensive disks (RAID) system [23], Microsoft Windows Azure cloud Storage [24] and as part of the Microsoft Real-Time Streaming Protocol [25], amongst others.

B. Low Density Parity Check Codes

The low density parity check (LDPC) codes as the name implies are very sparse random \mathbb{F}_2 linear codes, i.e. $\omega \ll k$, and the coding vector is selected randomly over \mathbb{F}_2 . The LDPC codes were invented by Gallager in 1962. The importance of these codes has largely remained dormant until the 1990s when Fountain codes were designed using irregular LDPC codes.

There are several decoding algorithms for LDPC, each for different channel model. For BEC, the message passing algorithm is used. The message passing algorithm is analogous to the back-substitution algorithm. A pseudocode of the message passing algorithm is given in [26, pp. 52].

C. Random Linear Codes

Random Linear (RL) codes, also known as Random Fountain codes and Random Linear Network Codes (RLNC), use coding coefficient \mathbf{g}_k randomly selected from \mathbb{F}_q . When the finite field from which the coding coefficients are randomly selected is large, then any k received coded packets are linearly independent with respect to each other with high probability. Analytical results on the performance of RL codes for different finite field size is given in [27].

The major drawback of RL codes is that it uses the computationally expensive Gaussian elimination method for decoding, which limits its implementation for larger input packet length [6], [7]. On the positive aspect, because of the random encoding procedure, RL code is in particular popular in network coding [28], as it can be implemented in a decentralized network system because of its random encoding process.

The sparsity of a coding vector is denoted by ω . A ω -sparse coding vector is one with $k - \omega$ zero coefficients and ω non-zero coefficients.

IV. FOUNTAIN CODES

We first give a formal definition of Fountain codes, adopted from [13]. An erasure code can be classified as a Fountain code, if it has the following traits.

- The number of coded packets which can be generated from a given set of input packets should be sufficiently large.
- Irrespective of which packets the client has received, the client should be able to decode the k input packets using any k linearly independent received packets.
- The encoding and decoding computation cost should be linear.

We now visit the development of various Fountain coding schemes.

A. Tornado Codes

Spurred by the popularity of the INTERNET in the 1990s and high decoding cost of RS codes, Luby *et al.* developed the Tornado codes in 1996 [29], [30]. Tornado codes have linear decoding cost of $n \log(\frac{1}{\epsilon})$, and transmission rate of 1.06. Software-based implementations of Tornado codes were shown to be about 100 times faster on small input packet size and about 10,000 times faster on larger input packet size than other software based implementation of RS codes [31].

Tornado codes are constructed using concatenation at different level. At the first layer, the input packets are XOR coded into coded packets of smaller length, these coded packets are then in turn XOR coded into coded packets at the second layer and so on. The last layer is coded using conventional code such as RS codes. Tornado codes are patented [32], [33].

Unfortunately Tornado codes are *fixed-rate* codes, i.e. once the encoder has chosen k and n based on initial channel erasure rate estimation, the encoder can generate only n codewords. Therefore if the average channel erasure rate is less than what the encoder initially estimated, then this would lead to redundant codewords at the decoder, whereas if the average erasure rate is higher than what was initially estimated, then that would lead the decoder unable to decode due to insufficient codewords. However for most internet applications and the wireless channel, the erasure rate has been shown to be stochastic in nature [1]. This motivates the design of rateless codes with linear decoding cost.

B. Luby-Transform and Raptor Codes

Armed with Tornado codes, Luby along with Goldin founded the Digital Fountain company to develop efficient erasure codes in 1998, drawing capital investment from Adobe, Cisco Systems and Sony Corporation [30]. The company and the codes the company developed are known as “Fountain,” based on what the codes achieve, generate virtually unlimited supply of codewords, analogous to a Fountain producing limitless drops of water. Just as an arbitrary collection of water drops will fill a glass of water and quench thirst, irrespective of which water drops had been collected. Collection of any n Fountain codewords will be sufficient for the decoder to decode the input packets.

The main idea behind Luby-Transform (LT) codes and Raptor codes is to design the *degree distribution* (such as the robust Soliton distribution, or one of its variant) of a coded packet. The *degree* of a coded packet, indicates the number of input packets used to generate the coded packet. LT codes and Raptor codes are irregular codes. LT coding is done in two steps, first the encoder randomly selects the degree, whose expected probability is dictated by the degree distribution. In the next step the encoder, randomly selects input packets, the number of input packets selected is given by the degree selected in the first step, and perform XOR addition of those input packets.

Decoding is performed using back-substitution. The decoder looks for those coded packet with one unknown input packets, and decode the unknown packet. It then substitutes this decoded packets in all the other coded packets which had been generated using this decoded packet as one of its constituent encoding packet. The decoder continues to repeat this process of decoding and substitution until it has not decoded all the k input packets. If it is unable to decode k input packets, then it requests for additional coded packets to be transmitted by the server.

Raptor (Rapid Tornado) codes is a special class of LT codes. The design of Raptor codes is motivated by the fact that due to the random nature of selecting the input packets, there is always a non-zero probability that some of the input packets may never be selected for coding in LT codes. To address this problem, in Raptor code the input symbols are first *precoded*, and then LT coding procedure takes place. Precoding can be thought of concatenation of input packets (c_1, c_2, \dots, c_k) and redundant packets (y_1, y_2, \dots, y_j) , given as $(c_1, \dots, c_k, y_1, \dots, y_j)$. The redundant packets can be generated by randomly coding the input packets using XOR addition. After the precoding the output packets are generated using LT coding procedure, whose input packets are given by the concatenation $(c_1, \dots, c_k, y_1, \dots, y_j)$.

Decoding Raptor codes is done using inactivation decoding. While a detailed illustrating example of decoding Raptor code is given in [13, 250-255], we here give the main idea behind inactivation using simpler example. Consider the coding coefficient matrix for the Raptor codes in Figure 2. Instead of performing Gaussian elimination on the complete matrix, inactivation algorithm first attempts to make the matrix sparse by back-substituting 1-sparse rows and the resulting 1-sparse rows. When no more 1-sparse coded packets are present, the algorithm decodes the resulting coded packets represented by a smaller but denser submatrix. Since Gaussian elimination runs on a much smaller submatrix, the resulting overall decoding complexity can be reduced.

C. LT and Raptor Codes Performance

Luby-Transform (LT) codes [34] and Raptor codes [9], along with Tornado codes, are classes of \mathbb{F}_2 linear codes broadly known as *Fountain codes*. Both LT and Raptor codes are rateless codes, with asymptotic decoding complexity of $\mathcal{O}(k \log(\frac{k}{\delta}))$ and $\mathcal{O}(k \log(\frac{1}{\epsilon}))$ respectively to deliver asymp-

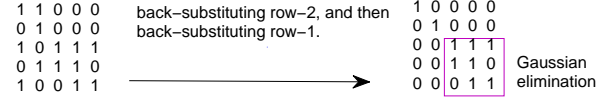


Fig. 2. An example to illustrate inactivation decoding.

totic optimality, where $1 - \delta$ is the probability that the LT decoder can recover the input packets from n codewords.

However for finite input packet length, particularly small values of k , LT and Raptor codes suffer from degrading transmission rate performance. Even for large input packet length of $k \approx 10,000$, LT codes have an overhead of 5% [8], while for packet length of $k = 65,536$, Raptor codes have an overhead of 3.8% [9]. Asymptotic rate optimality for LT and Raptor codes is therefore achieved only when the order of k is in 100,000's of packets.

In addition, both LT codes and Raptor codes are not systematic codes, therefore limiting its application for scenarios such as the index coding problem [17] and cooperative data exchange problem [11], where the decoder should be able to decode the input packets using a combination of subset of the input packets and coded packets. Similarly for Peer-to-Peer Content Distribution Network (P2P-CDN), a user may want to "preview" the content of the file before dedicating several hours to download the complete file, to verify that an incorrect file has not been uploaded by a dishonest user. The use of systematic codes is also a requirement in data storage, where the storage device should be able to recover an erased input packet, by using input packets and coded packet (also known as parity packets).

Hyttia *et al.* studied optimal degree distribution for LT over small input packet size of $k \leq 30$ [35]. The results concluded that even with optimized degree distribution, LT codes have a redundancy of approximately 40% for $k \leq 20$. These optimized degree distribution were calculated using a recursive equation, with exponential running time, making it computationally impractical to design optimal degree distribution for $k > 20$.

Patent description of LT codes are given in [36], [37], while those of Raptor codes are given in [38], [39].

D. Standardized Raptor Codes

Digital Fountain was later acquired by Qualcomm in 2009. During this time period, Digital Fountain went on to standardize the Raptor codes. These standardized and patented versions of Raptor codes are known as the Raptor 10 (R10) and Raptor Q (RQ) codes [13, Chapter 3] [40], and are systematic rateless codes designed to provide near-optimal transmission rates for finite length input packets.

The R10 codes generates some coded packets similar to RL codes over \mathbb{F}_2 , which are relatively denser. The remaining coded packets are sparsely coded based on the specified

TABLE I
PACKET RECEPTION STATUS EXAMPLE.

clients/packets	c_1	c_2	$c_1 \oplus c_2$
r_1	received	×	×
r_2	×	received	×
r_3	×	×	received

precoding and LT coding procedure. The performance of R10 codes can be approximated and upper bounded by the performance of RL codes over \mathbb{F}_2 . Analytical results of RL codes over \mathbb{F}_2 are given in [27], while the performances of sparse RL codes over \mathbb{F}_2 are reported in [41]. The RQ codes differs from R10, in that instead of generating some coded packets using RL codes over \mathbb{F}_2 , it uses a larger field size of \mathbb{F}_{256} . Similarly the performance of R10 codes are bounded by the performance of RL codes over \mathbb{F}_{256} . Performance evaluation of R10 and RQ codes can be found in [13, Chapter 3].

Such improvements in performance come at the tradeoff cost of using the inactivation decoding algorithm which uses combination of Gaussian elimination and belief-propagation decoding algorithm to decode the codewords. The decoding complexity of R10 and RQ codes is given as $\mathcal{O}(k^{1.5})$ [13, pp. 254-255] [40, pp. 8]. Practical implementation of R10 codes has shown that Gaussian elimination can account for up to 92% of the total decoding time even for a modestly large value of $k = 1024$, where each packet is 4 bytes long [42].

Even with the decoding complexity of $\mathcal{O}(k^{1.5})$, R10 codes can only support up to 8,192 input packets, while RQ codes can support up to 56,403 input packets. This limitation comes from the design of degree distribution of the codewords for R10 and RQ codes. Once the number of input packets exceeds these limits, the decoding failure probability gradually increases.

V. TRIANGULAR CODES

A new class of nonlinear erasure codes, which can be in practice be encoded and decoded using XOR addition operation despite being nonlinear, called Triangular codes, was proposed recently to further improve the performance while maintaining low decoding complexity. This class of codes uses a mix of finite field and real field operations for encoding, and the encoded packets can be decoding using back-substitution method involving only XOR addition. The tradeoff cost of Triangular codes is the additional redundant bits needed to pad to each packet. However, it has been demonstrated that a small number of additional bits is sufficient to lead the codes to a near-optimal performance of transmission rate while maintaining low decoding complexity for finite length input packets. A proof-of-concept for Triangular codes was first presented in [10]. A patent application for these codes has also been submitted [43]. Research and development for Triangular codes is currently an ongoing work.

$$\begin{array}{ccc}
 c_1 \oplus c_2 & & c_1 \oplus 2 \times c_2 \\
 \oplus \begin{array}{c} b_{1,1} \ b_{1,2} \ b_{1,3} \dots b_{1,B-1} \ b_{1,B} \\ b_{2,1} \ b_{2,2} \ b_{2,3} \dots b_{2,B-1} \ b_{2,B} \end{array} & & \oplus \begin{array}{c} b_{1,1} \ b_{1,2} \dots b_{1,B-1} \ b_{1,B} \\ b_{2,1} \ b_{2,2} \ b_{2,3} \dots b_{2,B} \ 0 \end{array}
 \end{array}$$

Fig. 3. An example to illustrate encoding process for $c_1 \oplus c_2$ and $c_{1,1} \oplus c_{2,0}$. The presence or absence of the added bit '0' at the head of $c_{1,1}$ does not effect the actual encoded bitstream, as long as the right most bits of the two packets are aligned.

A. Motivating Example

We illustrate the motivation of Triangular codes using an example. Consider for example, a server multicasting packet c_1 and c_2 to three clients. After the transmission on independent BEC, c_1 is received only by r_1 , and c_2 is received only by r_2 . Now the only linearly independent packet the server can transmit under the constraint of \mathbb{F}_2 linear coding is $c_1 \oplus c_2$. Assuming that the transmitted $c_1 \oplus c_2$ packet is received only by r_3 as shown in Table I, then with such packet reception status, there is no way the server can generate a \mathbb{F}_2 linear code which is linearly independent for all the three clients.

To go around the problem depicted in Table I, in our design, redundant '0' bits are selectively added at the head and tail of the input packets before performing XOR addition on packets. Assume that all packets c_i , $i \leq k$, are B -tuple, denoted as $c_i \triangleq (b_{i,1}, b_{i,2}, \dots, b_{i,B})$, $b_{i,j} \in \{0, 1\}$. Bit $b_{i,j}$ is read as the j^{th} bit of the i^{th} packet. Hence the bit pattern of c_1 with one redundant '0' bit added at the head of the packet is given by the tuple $(0, b_{1,1}, \dots, b_{1,B})$, and that of c_2 as $(b_{2,1}, \dots, b_{2,B}, 0)$. The packet header will include information about the number of redundant '0' bits added at the tail of each packet used to generate the coded packet.

The modified packets c_1 and c_2 can now represented as $c_{1,1}$ and $c_{2,0}$, where $c_{i,v}$ is read as the i^{th} packet with v -0 bits added at its head. The transmitted packet $c_{1,1} \oplus c_{2,0}$ will be linearly independent for all the three clients. An illustration of the encoding $c_1 \oplus c_2$ and $c_{1,1} \oplus c_{2,0}$, corresponding to $c_1 \oplus 2 \times c_2$ is shown in Figure 3.

If r_3 receives $c_{1,1} \oplus c_{2,0}$, then using packet $c_1 \oplus c_2$, it now has information about bit $b_{2,1}$, and $b_{1,1} \oplus b_{1,2}$. Using bit $b_{2,1}$ it can decode bit $b_{1,1}$ from $b_{1,1} \oplus b_{1,2}$. Bit $b_{1,1}$ is then substituted in $b_{1,1} \oplus b_{2,2}$, from the encoded packet $c_{1,1} \oplus c_{2,0}$, to obtain bit $b_{2,2}$, which can then be substituted in $b_{1,2} \oplus b_{2,2}$ to decode $b_{1,2}$. Therefore using this bit-by-bit simple back substitution method, R_3 can decode all the bits of packet c_1 and c_2 . Clients r_1 and r_2 can also similarly decode the unknown packet from $c_{1,1} \oplus c_{2,0}$.

B. Encoding and Decoding

The implementation of adding redundant '0' bits can be described by the real field multiplication of the bitstream tuple with 2^ℓ , $\ell \in \mathbb{N}_0$, where ℓ denotes the number of redundant '0' bits added at the tail of the packet. After the multiplication, with the right most bits (tail) of the concerned input packets aligned, the packets are XOR added to generate a coded

packet. To equalize the length of all packets for this finite field addition operation, redundant '0' bits are padded at the head of various packets, such that the total length of all packets used for encoding is equal. The encoded packet can be given as $c_1 \oplus (2 \times c_2)$, i.e. with coding coefficients given as $[1, 2]$. The encoding procedure for generating a coded packet κ from the set of packets \mathcal{P} using Triangular coding scheme can be given as,

$$\kappa = \bigoplus_{c_i \in \mathcal{P}} 2^\ell \times c_i.$$

The use of real field multiplication avoids the complicated finite field multiplication in the encoding. Real field multiplication can be easily accomplished by logical shift operation. While in the decoding, back-substitution can be used immediately on the set of encoded packets. The idea and design challenge of Triangular codes is to add redundant '0' bits to each packet such that there exists a row interchange permutation whereby these redundant '0' bits form a triangular pattern. If in addition to forming a triangular pattern, the coding vector also forms a full rank matrix, then the packets can be decoded using back-substitution only. Since in the coding coefficients, the unknown bits already form a Triangular matrix, the triangularization step of Gaussian elimination is no longer needed. The decoding complexity is therefore bounded as $\mathcal{O}(k^2)$ for each bit location using back-substitution. The complexity may be further reduced to $\mathcal{O}(k\omega)$, $\omega \leq k$, with a design of suitable sparse coding coefficients. As overhead of additional redundant bits are needed to pad to packets, the design challenge of Triangular codes is strike a balance between performance in retransmission rate and overhead for specific applications.

VI. FUTURE RESEARCH DIRECTIONS

In this section we discuss few open problems where research on erasure coding is ongoing, namely, decoding delay of erasure codes, pollution attack, and the index coding problem.

While the use of erasure code improves the bandwidth performance of a broadcast network, it has a disadvantage of incurring a decoding delay. For example, for a client which has packet c_1 and wants packets c_2 and c_3 , coded packets $c_2 \oplus c_3$ and $c_1 \oplus c_2$ are both linearly independent with respect to c_1 , however only the latter coded packet can be instantly decoded by the client. When considering RS and RL codes, a client need to have k linearly independent packets before it can start the decoding process. Designing an erasure coding scheme such that the time duration the client need to wait before it can decode the coded packet is an ongoing open research topic. We refer interested readers to [44] and references therein.

Another problem related to security aspects of erasure codes is that of pollution attack. If the client admits even a single malicious coded packet from a malicious user, then during the decoding process, all the decoded packets will be corrupted, and hence result in turning the correctly used coded packet as being useless, if no appropriate security mechanism is in place. Devising algorithmic approach to prevent such pollution

attack, and its corresponding computational complexity is also an ongoing research topic. We refer interested readers to [45] and references therein.

An overview of the index coding problem had been presented earlier in Section II-G.

VII. CONCLUSION

Traditional approaches to deal with system erasure are to use retransmission and replication techniques, which limits the reliability of the system, and adversely effects the throughput performance of the system. In this paper, we presented motivation for the use of erasure coding to improve the reliability and performance of data transmission and data storage system. However the principle disadvantage of using traditional erasure coding such as the Reed-Solomon coding is that such codes have high decoding complexity limiting its implementation, especially on battery and processor constrained devices such as smartphones.

In the last two decades, motivated by the exponential increase in the data traffic over the Internet, a series of Fountain codes - Tornado codes, LT codes, and Raptor codes - have been proposed, and patented, to address the decoding complexity of RS codes. Unfortunately Fountain codes can achieve linear decoding complexity only when the input packet length is asymptotically large. For smaller input packet length, Fountain codes suffer from degrading transmission rate.

To address this tradeoff, recently the Triangular codes had been proposed. Triangular codes are non-linear codes, where the encoder add redundant bits at the head and tail of a packet before performing \mathbb{F}_2 addition of the packet. The main idea behind Triangular codes is to code the packets such that apart from being linearly independent with high probability, such coded packets can be decoded using the back-substitution decoding algorithm.

With the increasing use of mobile devices for Internet access, erasure codes will certainly continue to play an important roles wireless data transmissions. More research works are warranted to further reduce the transmission overhead while maintaining near-optimal coding performance.

REFERENCES

- [1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level Measurements from an 802.11b Mesh Network," in *ACM SIGCOMM'04*, Portland, USA, Aug-Sep 2004.
- [2] E. Rozner, A. Padmanabha, Y. Mehta, L. Qiu, and M. Jafry, "ER: Efficient Retransmission Scheme For Wireless LANs," in *ACM CoNEXT'07*, New York, USA, Dec 2007.
- [3] K. Maraslis, P. Chatzimisios, and A. Boucouvalas, "IEEE 802.11aa: Improvements on video transmission over wireless LANs," in *IEEE ICC'12*, Ottawa, Canada, Jun 2012.
- [4] O. N. Vasilenko, *Number-Theoretic Algorithms in Cryptography*. American Mathematical Society, 2006.
- [5] P. Vingelmann, M. V. Pedersen, F. H. P. Fitzek, and J. Heide, "Multimedia Distribution using Network Coding on the iPhone Platform," in *ACM MCMC'10*, Firenze, Italy, Oct 2010.
- [6] H. Shojania and B. Li, "Random Network Coding on the iPhone: Fact or Fiction?" in *ACM NOSSDAV'09*, New York, USA, Jun 2009.
- [7] P. Vingelmann, P. Zanaty, F. H. P. Fitzek, and H. Charaf, "Implementation of Random Linear Network Coding on OpenGL-enabled Graphics Cards," in *IEEE EW'09*, Aalborg, Denmark, May 2009.

- [8] D. J. MacKay, "Fountain codes," *IEEE Proceedings Communications*, vol. 152, no. 6, pp. 1062–1068, Dec 2005.
- [9] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2251–2567, Jun 2006.
- [10] J. Qureshi, C. H. Foh, and J. Cai, "Optimal Solution for the Index Coding Problem Using Network Coding over GF(2)," in *IEEE SECON'12*, Seoul, Korea, June 2012.
- [11] S. E. Tajbakhsh, P. Sadeghi, and R. Shams, "A Generalized Model for Cost and Fairness Analysis in Coded Cooperative Data Exchange," in *IEEE NetCod'11*, Beijing, China, July 2011.
- [12] J. B. Fraleigh and R. A. Beauregard, *Linear Algebra*. Addison-Wesley Publishing Company, 1987.
- [13] A. Shokrollahi and M. Luby, "Raptor Codes," *Foundations and Trends in Communications and Information Theory*, vol. 6, no. 3-4, pp. 213–322, 2009.
- [14] G. A. Darmohray and E. D. Brooks, "Gaussian Techniques on Shared Memory Multiprocessor Computers," in *SIAM PPSC'87*, Los Angeles, USA, December 1987.
- [15] K. Park, J.-S. Park, and W. W. Ro, "On Improving Parallelized Network Coding with Dynamic Partitioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1547–1560, November 2010.
- [16] C. W. Sung, K. W. Shum, and H. Y. Kwan, "On the Sparsity of a Linear Network Code for Broadcast Systems with Feedback," in *IEEE NetCod'11*, Beijing, China, July 2011.
- [17] M. Chaudhry and A. Sprintson, "Efficient Algorithms for Index Coding," in *IEEE INFOCOM Workshop'08*, Phoenix, USA, April 2008.
- [18] P. A. Chou and Y. Wu, "Network Coding for the Internet and Wireless Networks," *IEEE Signal Processing Magazine*, pp. 77–85, September 2007.
- [19] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun 1960.
- [20] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr 1997.
- [21] S. B. Wicker and M. Bartz, *Reed-Solomon Codes and Their Applications*. IEEE Press, 1994.
- [22] L. Mirsky, *An Introduction to Linear Algebra*. Dover Publications, 1982.
- [23] J. S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems," *Wiley Software: Practice and Experience*, vol. 27, no. 9, pp. 995–1012, Sept 1997.
- [24] B. Calder et al., "Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency," in *ACM SOSR'11*, Cascais, Portugal, Oct 2011.
- [25] [MS-RTSP]: Real-Time Streaming Protocol (RTSP) Windows Media Extensions, Oct 2012. [Online]. Available: <http://msdn.microsoft.com/en-us/library/cc245238.aspx>
- [26] S. J. Johnson, *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. Cambridge University Press, 2010.
- [27] O. T.-Cruces, J. B.-Ordinas, and M. Fiore, "Exact Decoding Probability Under Random Linear Network Coding," *IEEE Communications Letters*, vol. 15, no. 1, pp. 67–69, Jan 2011.
- [28] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network Coding Theory," *Foundation and Trends in Communications and Information Theory*, vol. 2, no. 4-5, pp. 241–381, 2005.
- [29] M. G. Luby, "Practical Loss-Resilient Codes," in *ACM STOC'97*, El Paso, USA, May 1997.
- [30] S. Robinson, "Beyond Reed-Solomon: New Codes for Internet Multicasting Drive Silicon Valley Start-up," *SIAM News*, vol. 35, no. 4, pp. 1–3, May 2002.
- [31] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *ACM SIGCOMM'98*, Vancouver, Canada, Aug-Sept 1998.
- [32] M. Luby, M. A. Shokrollahi, V. Stemann, M. D. Mitzenmacher, and D. A. Spielman, "Irregularly graphed encoding technique," Patent US 6,081,919, Jun. 27, 2000.
- [33] —, "Loss resilient decoding technique," Patent US 6,073,250, Jun. 6, 2000.
- [34] M. Luby, "LT Codes," in *IEEE FOCS'02*, Vancouver, Canada, Nov 2002.
- [35] E. Hyttia, T. Tirronen, and J. Virtamo, "Optimal Degree Distribution for LT Codes with Small Message Length," in *IEEE INFOCOM'07*, Anchorage, USA, May 2007.
- [36] M. Luby, "Information additive code generator and decoder for communication systems," Patent US 6,307,487, Oct. 23, 2001.
- [37] —, "Information additive code generator and decoder for communication systems," Patent US 6,373,406, April 16, 2002.
- [38] A. Shokrollahi, S. Lassen, and M. Luby, "Multi-stage code generator and decoder for communication systems," Patent US 2003/0058958 A1, Mar. 27, 2003.
- [39] A. Shokrollahi and M. G. Luby, "Systematic encoding and decoding of chain reaction codes," Patent US 6,909,383 B2, Jun. 21, 2005.
- [40] RaptorQ Technical Overview, QUALCOMM Incorporated, Oct 2010. [Online]. Available: www.qualcomm.com/media/documents/raptorq-technical-overview
- [41] C. Studholme and I. F. Blake, "Random Matrices and Codes for the Erasure Channel," *Algorithmica*, vol. 56, no. 4, pp. 605–620, Apr 2010.
- [42] T. Mladenov, S. Nooshabadi, and K. Kim, "Implementation and Evaluation of Raptor Codes on Embedded Systems," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1678–1691, Dec 2011.
- [43] J. Qureshi, C. H. Foh, and J. Cai, "Triangular network coding," Patent US Application 61/807,557, Nanyang Technological University, Filed on 02/April/2013.
- [44] L. Keller, E. Drinea, and C. Fragouli, "Online Broadcasting with Network Coding," in *IEEE NetCod'08*, Hong Kong, China, January 2008.
- [45] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks," in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, April 2009.